Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction

Cheng Niu, Wei Li, and Rohini K. Srihari
Cymfony Inc.
600 Essjay Road, Williamsville, NY 14221, USA.
{cniu, wei, rohini}@cymfony.com

Abstract

It is fairly common that different people are associated with the same name. In tracking person entities in a large document pool, it is important to determine whether multiple mentions of the same name across documents refer to the same entity or not. Previous approach to this problem involves measuring context similarity only based on co-occurring words. This paper presents a new algorithm using information extraction support in addition to co-occurring words. A learning scheme with minimal supervision is developed within the Bayesian framework. Maximum entropy modeling is then used to represent the probability distribution of context similarities based on heterogeneous features. Statistical annealing is applied to derive the final entity coreference chains by globally fitting the pairwise context similarities. Benchmarking shows that our new approach significantly outperforms the existing algorithm by 25 percentage points in overall F-measure.

1 Introduction

Cross document name disambiguation is required for various tasks of knowledge discovery from textual documents, such as entity tracking, link discovery, information fusion and event tracking. This task is part of the co-reference task: if two mentions of the same name refer to same (different) entities, by definition, they should (should not) be co-referenced. As far as names are concerned, co-reference consists of two sub-tasks: (i) name disambiguation to handle the problem of different entities happening to use the same name; (ii) alias association to handle the problem of the same entity using multiple names (aliases). Message Understanding Conference (MUC) community has established within-document co-reference standards [MUC-7 1998]. Compared with within-document name disambiguation which can leverage highly reliable discourse heuristics such as one sense per discourse [Gale et al 1992], cross-document name disambiguation is a much harder problem.

Among major categories of named entities (NEs, which in this paper refer to entity names, excluding the MUC time and numerical NEs), company and product names are often trademarked or uniquely registered, and hence less subject to name ambiguity. This paper focuses on cross-document disambiguation of person names.

Previous research for cross-document name disambiguation applies vector space model (VSM) for context similarity, only using co-occurring words [Bagga & Baldwin 1998]. A pre-defined threshold decides whether two context vectors are different enough to represent two different entities. This approach faces two challenges: i) it is difficult to incorporate natural language processing (NLP) results in the VSM framework; [1] ii) the algorithm focuses on the local pairwise context similarity, and neglects the global correlation in the data: this may cause inconsistent results, and hurts the performance.

This paper presents a new algorithm that addresses these problems. A learning scheme with minimal supervision is developed within the Bayesian framework. Maximum entropy modeling is then used to represent the probability distribution of context similarities based on heterogeneous features covering both co-occurring words and natural language information extraction (IE) results. Statistical annealing is used to derive the final entity co-reference chains by globally fitting the pairwise context similarities.

Both the previous algorithm and our new algorithm are implemented, benchmarked and

[1] Based on our experiment, only using co-occurring words often cannot fulfill the name disambiguation task. For example, the above algorithm identifies the mentions of *Bill Clinton* as referring to two different persons, one represents his role as U. S. president, and the other is strongly associated with the scandal, although in both mention clusters, Bill Clinton has been mentioned as U.S. president. Proper name disambiguation calls for NLP/IE support which may have extracted the key person's identification information from the textual documents.

compared. Significant performance enhancement up to 25 percentage points in overall F-measure is observed with the new approach. The generality of this algorithm ensures that this approach is also applicable to other categories of NEs.

The remaining part of the paper is structured as follows. Section 2 presents the algorithm design and task definition. The name disambiguation algorithm is described in Sections 3, 4 and 5, corresponding to the three key aspects of the algorithm, i.e. minimally supervised learning scheme, maximum entropy modeling and annealing-based optimization. Benchmarks are shown in Section 6, followed by Conclusion in Section 7.

2 Task Definition and Algorithm Design

Given n name mentions, we first introduce the following symbols. C_i refers to the context of the i-th mention. P_i refers to the entity for the i-th mention. $Name_i$ refers to the name string of the i-th mention. $CS_{i,j}$ refers to the context similarity between the i-th mention and the j-th mention, which is a subset of the predefined context similarity features. f_α refers to the α-th predefined context similarity feature. So $CS_{i,j}$ takes the form of $\{f_\alpha\}$.

The name disambiguation task is defined as hard clustering of the multiple mentions of the same name. Its final solution is represented as $\{K, M\}$ where K refers to the number of distinct entities, and M represents the many-to-one mapping (from mentions to a cluster) such that $\mathrm{M(i)=j, i\in[1,n], j\in[1,K]}$.

One way of combining natural language IE results with traditional co-occurring words is to design a new context representation scheme and then define the context similarity measure based on the new scheme. The challenge to this approach lies in the lack of a proper weighting scheme for these high-dimensional heterogeneous features. In our research, the algorithm directly models the pairwise context similarity.

For any given context pair, a set of predefined context similarity features are defined. Then with n mentions of a same name, $\frac{n(n-1)}{2}$ context similarities $CS_{i,j}\ (i\in[1,n], j\in[1,i))$ are computed. The name disambiguation task is formulated as searching for $\{K, M\}$ which maximizes the following conditional probability: $\Pr(\{K,M\}|\{CS_{i,j}\})\quad (i\in[1,n], j\in[1,i))$

Based on Bayesian Equity, this is equivalent to maximizing the following joint probability

$$\begin{aligned}&\Pr(\{K,M\},\{CS_{i,j}\})\quad (i\in[1,n], j\in[1,i))\\&=\Pr(\{CS_{i,j}\}|\{K,M\})\Pr(\{K,M\})\qquad (1)\\&\approx\prod_{\substack{i=1,N\\ j=1,i-1}}\Pr(CS_{i,j}|\{K,M\})\Pr(\{K,M\})\end{aligned}$$

Eq. (1) contains a prior probability distribution of name disambiguation $\Pr(\{K,M\})$. Because there is no prior knowledge available about what solution is preferred, it is reasonable to take an equal distribution as the prior probability distribution. So the name disambiguation is equivalent to searching for $\{K,M\}$ which maximizes Expression (2).

$$\prod_{\substack{i=1,N\\ j=1,i-1}}\Pr(CS_{i,j}|\{K,M\})\qquad (2)$$

where

$$\Pr(CS_{i,j}|\{K,M\})=\begin{cases}\Pr(CS_{i,j}|P_i=P_j), \text{if } \mathrm{M(i)=M(j)}\\ \Pr(CS_{i,j}|P_i\neq P_j), \text{otherwise}\end{cases}\qquad (3)$$

To learn the conditional probabilities $\Pr(CS_{i,j}\mid P_i=P_j)$ and $\Pr(CS_{i,j}\mid P_i\neq P_j)$ in Eq. (3), we use a machine learning scheme which only requires minimal supervision. Within this scheme, maximum entropy modeling is used to combine heterogeneous context features. With the learned conditional probabilities in Eq. (3), for a given $\{K,M\}$ candidate, we can compute the conditional probability of Expression (2). In the final step, optimization is performed to search for $\{K,M\}$ that maximizes the value of Expression (2).

To summarize, there are three key elements in this learning scheme: (i) the use of automatically constructed corpora to estimate conditional probabilities of Eq. (3); (ii) maximum entropy modeling for combining heterogeneous context similarity features; and (iii) statistical annealing for optimization.

3 Learning Using Automatically Constructed Corpora

This section presents our machine learning scheme to estimate the conditional probabilities $\Pr(CS_{i,j}\mid P_i=P_j)$ and $\Pr(CS_{i,j}\mid P_i\neq P_j)$ in Eq. (3). Considering $CS_{i,j}$ is in the form of $\{f_\alpha\}$, we re-formulate the two conditional probabilities as

$\Pr(\{f_\alpha\} | P_i = P_j)$ and $\Pr(\{f_\alpha\} | P_i \neq P_j)$.

The learning scheme makes use of automatically constructed large corpora. The rationale is illustrated in the figure below. The symbol + represents a positive instance, namely, a mention pair that refers to the same entity. The symbol – represents a negative instance, i.e. a mention pair that refers to different entities.

Corpus I	Corpus II
+++++---++++++	----------
+-----+++--+++++	--+----------
++++++++++--++	----------+------
+++++++---++++	----------
+++----++++++++	--------+----------

As shown in the figure, two training corpora are automatically constructed. Corpus I contains mention pairs of the same names; these are the most frequently mentioned names in the document pool. It is observed that frequently mentioned person names in the news domain are fairly unambiguous, hence enabling the corpus to contain *mainly* positive instances.[2] Corpus II contains mention pairs of different person names, these pairs *overwhelmingly* correspond to negative instances (with statistically negligible exceptions). Thus, typical patterns of negative instances can be learned from Corpus II. We use these patterns to filter away the negative instances in Corpus I. The purified Corpus I can then be used to learn patterns for positive instances. The algorithm is formulated as follows.

Following the observation that different names usually refer to different entities, it is safe to derive Eq. (4).

$$\Pr(\{f_\alpha\} | P_1 \neq P_2) = \Pr(\{f_\alpha\} | name_1 \neq name_2) \quad (4)$$

For $\Pr(\{f_\alpha\} | P_1 = P_2)$, we can derive the following relation (Eq. 5):

$$\begin{aligned} &\Pr(\{f_\alpha\} | name_1 = name_2) \\ &= [\Pr(\{f_\alpha\} | P_1 = P_2) \\ &\quad * \Pr(P_1 = P_2 | name_1 = name_2)] \\ &+ [\Pr(\{f_\alpha\} | P_1 \neq P_2) \\ &\quad * (1 - \Pr(P_1 = P_2 | name_1 = name_2))] \end{aligned} \quad (5)$$

So $\Pr(\{f_\alpha\} | P_1 = P_2)$ can be determined if $\Pr(\{f_\alpha\} | name(P_1) = name(P_2))$, $\Pr(\{f_\alpha\} | name(P_1) \neq name(P_2))$, and $\Pr(P_1 = P_2 | name(P_1) = name(P_2))$ are all known.

By using Corpus I and Corpus II to estimate the above three probabilities, we achieve Eq. (6.1) and Eq. (6.2)

$$\Pr(\{f_\alpha\} | P_1 = P_2) = \frac{\Pr_{\mathrm{I}}^{\mathrm{maxEnt}}(\{f_\alpha\}) - \Pr_{\mathrm{II}}^{\mathrm{maxEnt}}(\{f_\alpha\}) * (1 - X)}{X}. \quad (6.1)$$

$$\Pr(\{f_\alpha\} | P_1 \neq P_2) = \Pr_{\mathrm{II}}^{\mathrm{maxEnt}}(\{f_\alpha\}) \quad (6.2)$$

where $\Pr_{\mathrm{I}}^{\mathrm{maxEnt}}(\{f_\alpha\})$ denotes the maximum entropy model of $\Pr(\{f_\alpha\} | name(P_1) = name(P_2))$ using Corpus I, $\Pr_{\mathrm{II}}^{\mathrm{maxEnt}}(\{f_\alpha\})$ denotes the maximum entropy model of $\Pr(\{f_\alpha\} | name(P_1) \neq name(P_2))$ using Corpus II, and X stands for the Maximum Likelihood Estimation (MLE) of $\Pr(P_1 = P_2 | name(P_1) = name(P_2))$ using Corpus I. Maximum entropy modeling is used here due to its strength of combining heterogeneous features.

It is worth noting that $\Pr_{\mathrm{I}}^{\mathrm{maxEnt}}(\{f_\alpha\})$ and $\Pr_{\mathrm{II}}^{\mathrm{maxEnt}}(\{f_\alpha\})$ can be automatically computed using Corpus I and Corpus II. Only X requires manual truthing. Because X is context independent, the required truthing is very limited (in our experiment, only 100 truthed mention pairs were used). The details of corpus construction and truthing will be presented in the next section.

4 Maximum Entropy Modeling

This section presents the definition of context similarity features $\{f_\alpha\}$, and how to estimate the maximum entropy model of $\Pr_{\mathrm{I}}^{\mathrm{maxEnt}}(\{f_\alpha\})$ and $\Pr_{\mathrm{II}}^{\mathrm{maxEnt}}(\{f_\alpha\})$.

First, we describe how Corpus I and Corpus II are constructed. Before the person name

[2] Based on our data analysis, there is no observable difference in linguistic expressions involving frequently mentioned vs. occasionally occurring person names. Therefore, the use of frequently mentioned names in the corpus construction process does not affect the effectiveness of the learned model to be applicable to all the person names in general.

disambiguation learning starts, a large pool of textual documents are processed by an IE engine *InfoXtract* [Srihari *et al* 2003]. The InfoXtract engine contains a named entity tagger, an aliasing module, a parser and an entity relationship extractor. In our experiments, we used ~350,000 AP and WSJ news articles (a total of ~170 million words) from the TIPSTER collection. All the documents and the IE results are stored into an IE Repository. The top 5,000 most frequently mentioned multi-token person names are retrieved from the repository. For each name, all the contexts are retrieved while the context is defined as containing three categories of features:

(i) The surface string sequence centering around a key person name (or its aliases as identified by the aliasing module) within a predefined window size equal to 50 tokens to both sides of the key name.

(ii) The automatically tagged entity names co occurring with the key name (or its aliases) within the same predefined window as in (i).

(iii) The automatically extracted relationships associated with the key name (or its aliases). The relationships being utilized are listed below:

Age, Where-from, Affiliation, Position, Leader-of, Owner-of, Has-Boss, Boss-of, Spouse-of, Has-Parent, Parent-of, Has-Teacher, Teacher-of, Sibling-of, Friend-of, Colleague-of, Associated-Entity, Title, Address, Birth-Place, Birth-Time, Death-Time, Education, Degree, Descriptor, Modifier, Phone, Email, Fax.

A recent manual benchmarking of the InfoXtract relationship extraction in the news domain is 86% precision and 67% recall (75% F-measure).

To construct Corpus I, a person name is randomly selected from the list of the top 5,000 frequently mentioned multi-token names. For each selected name, a pair of contexts are extracted, and inserted into Corpus I. This process repeats until 10,000 pairs of contexts are selected.

It is observed that, in the news domain, the top frequently occurring multi-token names are highly unambiguous. For example, *Bill Clinton* exclusively stands for the previous U.S. president although in real life, although many other people may also share this name. Based on manually checking 100 sample pairs in Corpus I, we have $X = \Pr_I(P_1 = P_2) \approx 0.95$, which means for the 100 sample pairs mentioning the same person name, only 5 pairs are found to refer to different person entities. Note that the value of $1 - X$ represents the estimation of the noise in Corpus I, which is used in Eq (6.1) to correct the bias caused by the noise in the corpus.

To construct Corpus II, two person names are randomly selected from the same name list. Then a context for each of the two names is extracted, and this context pair is inserted into Corpus II. This process repeats until 10,000 pairs of contexts are selected.

Based on the above three categories of context features, four context similarity features are defined:

(1) VSM-based context similarity using co-occurring words

The surface string sequence centering around the key name is represented as a vector, and the word *i* in context *j* is weighted as follows.

$$weight(i, j) = tf(i, j) * \log \frac{D}{df(i)} \qquad (7)$$

where $tf(i, j)$ is the frequency of word *i* in the *j-th* surface string sequence; D is the number of documents in the pool; and $df(i)$ is the number of documents containing the word *i*. Then, the cosine of the angle between the two resulting vectors is used as the context similarity measure.

(2) Co-occurring NE Similarity

The latent semantic analysis (LSA) [Deerwester et al 1990] is used to compute the co-occurring NE similarities. LSA is a technique to uncover the underlining semantics based on co-occurrence data. The first step of LSA is to construct word-vs.-document co-occurrence table. We use 100,000 documents from the TIPSTER corpus, and select the following types of top *n* most frequently mentioned words as base words:

top 20,000 common nouns
top 10,000 verbs
top 10,000 adjectives
top 2,000 adverbs
top 10,000 person names
top 15,000 organization names
top 6,000 location names
top 5,000 product names

Then, a word-vs.-document co-occurrence table *Matrix* is built so that

$Matrix_{ij} = tf(i,j) * \log\frac{D}{df(i)}$. The second step of LSA is to perform singular value decomposition (SVD) on the co-occurrence matrix. SVD yields the following *Matrix* decomposition:

$$Matrix = T_0 S_0 D_0^T \quad (8)$$

where T and D are orthogonal matrices (the row vector is called singular vectors), and S is a diagonal matrix with the diagonal elements (called singular values) sorted decreasingly.

The key idea of LSA is to reduce noise or insignificant association patterns by filtering the insignificant components uncovered by SVD. This is done by keeping only top *k* singular values. In our experiment, *k* is set to 200, following the practice reported in [Deerwester et al. 1990] and [Landauer & Dumais, 1997]. This procedure yields the following approximation to the co-occurrence matrix:

$$Matrix \approx TSD^T \quad (9)$$

where S is attained from S_0 by deleting non-top *k* elements, and T (D) is obtained from T_0 (D_0) by deleting the corresponding columns.

It is believed that the approximate matrix is more proper to induce underlining semantics than the original one. In the framework of LSA, the co-occurring NE similarities are computed as follows: suppose the first context in the pair contains NEs $\{t_{0i}\}$, and the second context in the pair contains NEs $\{t_{1i}\}$. Then the similarity is computed as $S = \frac{\sum w_{0i} T_{t_{0i}} \sum w_{1i} T_{t_{1i}}}{\left\|\sum w_{0i} T_{t_{0i}}\right\| \left\|\sum w_{1i} T_{t_{1i}}\right\|}$ where w_{0i} and w_{1i} are term weights defined in Eq (7).

(3) Relationship Similarity

We define four different similarity values based on entity relationship sharing: (i) *sharing no common relationships,* (ii) *relationship conflicts only,* (iii) *relationship with consistence* and *conflicts,* and (iv) *relationship with consistence only*. The consistency checking between extracted relationships is supported by the InfoXtract number normalization and time normalization as well as entity aliasing procudures.

(4) Detailed Relationship Similarity

For each relationship type, four different similarity values are defined based on sharing of that specific relationship *i*: (i) *no sharing of relationship i,* (ii) *conflicts for relationship i,* (iii) *consistence and conflicts for relationship i,* and (iv) *consistence for relationship i.*

To facilitate the maximum entropy modeling in the later stage, the values of the first and second categories of similarity measures are discretized into integers. The number of integers being used may impact the final performance of the system. If the number is too small, significant information may be lost during the discretization process. On the other hand, if the number is too large, the training data may become too sparse. We trained a conditional maximum entropy model to disambiguate context pairs between Corpus I and Corpus II. The performance of this model is used to select the optimal number of integers. There is no significant performance change when the integer number is within the range of [5,30], with 12 as the optimal number.

Now the context similarity for a context pair is a vector of similarity features, e.g.

{VSM_Similairty_equal_to_2,
NE_Similarity_equal_to_1,
Relationship_Conflicts_only,
No_Sharing_for_Age,
Conflict_for_Affiliation}.

Besides the four categories of basic context similarity features defined above, we define induced context similarity features by combining basic context similarity features using the logical *AND* operator. With induced features, the context similarity vector in the previous example is represented as

{VSM_Similairty_equal_to_2,
NE_Similarity_equal_to_1,
Relationship_Conflicts_only,
No_Sharing_for_Age,
Conflict_for_Affiliation,
[VSM_Similairty_equal_to_2 *and*
NE_Similarity_equal_to_1],
[VSM_Similairty=2 *and*
Relationship_Conflicts_only],
……
[VSM_Similairty_equal_to_2 *and*
NE_Similarity_equal_to_1 *and*
Relationship_Conflicts_only *and*
No_Sharing_for_Age *and*
Conflict_for_Affiliation]
}.

The induced features provide direct and fine-grained information, but suffer from less sampling space. Combining basic features and induced

features under a smoothing scheme, maximum entropy modeling may achieve optimal performance.

Now the maximum entropy modeling can be formulated as follows: given a pairwise context similarity vector $\{f_\alpha\}$ the probability of $\{f_\alpha\}$ is given as

$$\Pr^{\text{maxEnt}}(\{f_\alpha\}) = \frac{1}{Z}\prod_{f\in\{f_\alpha\}} w_f \qquad (10)$$

where Z is the normalization factor, w_f is the weight associated with feature f. The Iterative Scaling algorithm combined with Monte Carlo simulation [Pietra, Pietra & Lafferty 1995] is used to train the weights in this generative model. Unlike the commonly used conditional maximum entropy modeling which approximates the feature configuration space as the training corpus [Ratnaparkhi 1998], Monte Carlo techniques are required in the generative modeling to simulate the possible feature configurations. The exponential prior smoothing scheme [Goodman 2003] is adopted. The same training procedure is performed using Corpus I and Corpus II to estimate $\Pr_{\text{I}}^{\text{maxEnt}}(\{f_i\})$ and $\Pr_{\text{II}}^{\text{maxEnt}}(\{f_i\})$ respectively.

5 Annealing-based Optimization

With the maximum entropy modeling presented in the last section, for a given name disambiguation candidate solution $\{K, M\}$, we can compute the conditional probability of Expression (2). Statistical annealing [Neal 1993]-based optimization is used to search for $\{K, M\}$ which maximizes Expression (2).

The optimization process consists of two steps. First, a local optimal solution $\{K, M\}_0$ is computed by a greedy algorithm. Then by setting $\{K, M\}_0$ as the initial state, statistical annealing is applied to search for the global optimal solution.

Given n same name mentions, assuming the input of $\frac{n(n-1)}{2}$ probabilities $\Pr(CS_{i,j}|P_i = P_j)$ and $\frac{n(n-1)}{2}$ probabilities $\Pr(CS_{i,j}|P_i \neq P_j)$, the greedy algorithm performs as follows:

1. Set the initial state $\{K, M\}$as $K = n$, and $M(i) = i,\ \mathrm{i} \in [1, \mathrm{n}]$;
2. Sort $\Pr(CS_{i,j}|P_i = P_j)$ in decreasing order;
3. Scan the sorted probabilities one by one. If the current probability is $\Pr(CS_{i,j}|P_i = P_j)$, $M(i) \neq M(j)$, and there exist no such l and m that $M(l) = M(i), M(m) = M(j)$ and $\Pr(CS_{i,j}|P_i = P_j) < \Pr(CS_{l,m}|P_l \neq P_m)$ then update $\{K, M\}$ by merging cluster $M(i)$ and $M(j)$.
4. Output $\{K, M\}$ as a local optimal solution.

Using the output $\{K, M\}_0$ of the greedy algorithm as the initial state, the statistical annealing is described using the following pseudo-code:

```
Set {K,M} = {K,M}_0 ;
for(β = β_0 ; β < β_final ; β* = 1.01)
{
    iterate pre-defined number of times
    {
      set {K,M}_1 = {K,M};
      update {K,M}_1 by randomly changing
      the number of clusters K and the
      content of each cluster.
                    ∏_{i=1,N; j=1,i-1} Pr(CS_{i,j} | {K,M}_1)
      set x = ------------------------------------------
                    ∏_{i=1,N; j=1,i-1} Pr(CS_{i,j} | {K,M})
      if(x>=1)
      {
        set {K,M} = {K,M}_1
      }
      else
      {
        set {K,M} = {K,M}_1 with probability
          x^β .
      }
                    ∏_{i=1,N; j=1,i-1} Pr(CS_{i,j} | {K,M})
      if ------------------------------------------ > 1
                    ∏_{i=1,N; j=1,i-1} Pr(CS_{i,j} | {K,M}_0)
      set {K,M}_0 = {K,M}
    }
}
output {K,M}_0 as the optimal state.
```

6 Benchmarking

To evaluate the effectiveness of our new algorithm, we implemented the previous algorithm described in [Bagga & Baldwin 1998] as our

baseline. The threshold is selected as 0.19 by optimizing the pairwise disambiguation accuracy using the 80 truthed mention pairs of "*John Smith*". To clearly benchmark the performance enhancement from IE support, we also implemented a system using the same weakly supervised learning scheme but only VSM-based similarity as the pairwise context similarity measure. We benchmarked the three systems for comparison. The following three scoring measures are implemented.

(1) Precision (P):

$$P = \frac{1}{N}\sum_{i}\frac{\#\text{ of correct mentions in the output cluster of i}}{\#\text{ of mentions in the output cluster of i}}$$

(2) Recall (R):

$$P = \frac{1}{N}\sum_{i}\frac{\#\text{ of correct mentions in the output cluster of i}}{\#\text{ of mentions in the key cluster of i}}$$

(3) F-measure (F):

$$F = \frac{2P*R}{P+R}$$

The name co-reference precision and recall used here is adopted from the B_CUBED scoring scheme used in [Bagga & Baldwin 1998], which is believed to be an appropriate benchmarking standard for this task.

Traditional benchmarking requires manually dividing person name mentions into clusters, which is labor intensive and difficult to scale up. In our experiments, an automatic corpus construction scheme is used in order to perform large-scale testing for reliable benchmarks.

The intuition is that in the general news domain, some multi-token names associated with mass media celebrities is highly unambiguous. For example, "Bill Gates", "Bill Clinton", etc. mentioned in the news almost always refer to unique entities. Therefore, we can retrieve contexts of these unambiguous names, and mix them together. The name disambiguation algorithm should recognize mentions of the same name. The capability of recognizing mentions of an unambiguous name is equivalent to the capability of disambiguating ambiguous names.

For the purpose of benchmarking, we automatically construct eight testing datasets (Testing Corpus I), listed in Table 1.

Table 1. Constructed Testing Corpus I

Name	**# of Mentions**	
	Set 1a	**Set 1b**
Mikhail S. Gorbachev	20	50
Dick Cheney	20	10
Dalai Lama	20	10
Bill Clinton	20	10
	Set 2a	**Set 2b**
Bob Dole	20	50
Hun Sen	20	10
Javier Perez de Cuellar	20	10
Kim Young Sam	20	10
	Set 3a	**Set 3b**
Jiang Qing	20	10
Ingrid Bergman	20	10
Margaret Thatcher	20	50
Aung San Suu Kyi	20	10
	Set 4a	**Set 4b**
Bill Gates	20	10
Jiang Zemin	20	10
Boris Yeltsin	20	50
Kim Il Sung	20	10

Table 2. Testing Corpus I Benchmarking

	P	**R**	**F**	**P**	**R**	**F**
	Set 1a			**Set 1b**		
Baseline	0.79	0.37	0.58	0.78	0.34	0.56
VSMOnly	0.86	0.33	0.60	0.78	0.23	0.51
Full	0.98	0.75	0.86	0.90	0.79	0.85
	Set 2a			**Set 2b**		
Baseline	0.82	0.58	0.70	0.94	0.50	0.72
VSMOnly	0.90	0.54	0.72	0.98	0.45	0.71
Full	0.93	0.84	0.88	1.00	0.93	0.96
	Set 3a			**Set 3b**		
Baseline	0.84	0.69	0.77	0.80	0.34	0.57
VSMOnly	0.95	0.72	0.83	0.93	0.29	0.61
Full	0.95	0.86	0.90	0.98	0.57	0.77
	Set 4a			**Set 4b**		
Baseline	0.88	0.74	0.81	0.80	0.49	0.64
VSMOnly	0.93	0.77	0.85	0.88	0.42	0.65
Full	0.95	0.93	0.94	0.98	0.84	0.91
Overall	**P**		**R**		**F**	
Baseline	0.83		0.51		**0.63**	
VSMOnly	0.90		0.47		**0.69**	
Full	0.96		0.82		**0.88**	

Table 2 shows the benchmarks for each dataset, using the three measures just defined. The new algorithm when only using VSM-based similarity (*VSMOnly*) outperforms the existing algorithm (Baseline) by 5%. The new algorithm using the full context similarity measures including IE features (*Full*) significantly outperforms the existing algorithm (*Baseline*) in every test: the overall F-

measure jumps from 64% to 88%, with 25 percentage point enhancement. This performance breakthrough is mainly due to the additional support from IE, in addition to the optimization method used in our algorithm.

We have also manually truthed an additional testing corpus of two datasets containing mentions associated with the same name (Testing Corpus II). Truthed Dataset 5a contains 25 mentions of *Peter Sutherland* and Truthed Dataset 5b contains 68 mentions of *John Smith*. *John Smith* is a highly ambiguous name. With its 68 mentions, they represent totally 29 different entities. On the other hand, all the mentions of *Peter Sutherland* are found to refer to the same person. The benchmark using this corpus is shown below.

Table 3. Testing Corpus II Benchmarking

	P	**R**	**F**	**P**	**R**	**F**
	Set 5a			**Set 5b**		
Baseline	0.96	0.92	0.94	0.62	0.57	0.60
VSMOnly	0.96	0.92	0.94	0.75	0.51	0.63
Full	1.00	0.92	0.96	0.90	0.81	0.85

Based on these benchmarks, using either manually truthed corpora or automatically constructed corpora, using either ambiguous corpora or unambiguous corpora, our algorithm consistently and significantly outperforms the existing algorithm. In particular, our system achieves a very high precision (0.96 precision). This shows the effective use of IE results which provide much more fine-grained evidence than co-occurring words. It is interesting to note that the recall enhancement is greater than the precision enhancement (0.31 recall enhancement vs. 0.13 precision enhancement). This demonstrates the complementary nature between evidence from the co-occurring words and the evidence carried by IE results. The system recall can be further improved once the recall of the currently precision-oriented IE engine is enhanced over time.

7 Conclusion

We have presented a new person name disambiguation algorithm which demonstrates a successful use of natural language IE support in performance enhancement. Our algorithm is benchmarked to outperform the previous algorithm by 25 percentage points in overall F-measure, where the effective use of IE contributes to 20 percentage points. The core of this algorithm is a learning system trained on automatically constructed large corpora, only requiring minimal supervision in estimating a context-independent probability.

8 Acknowledgements

This work was partly supported by a grant from the Air Force Research Laboratory's Information Directorate (AFRL/IF), Rome, NY, under contract F30602-03-C-0170. The authors wish to thank Carrie Pine of AFRL for supporting and reviewing this work.

References

Bagga, A., and B. Baldwin. 1998. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In *Proceedings of COLING-ACL'98*.

Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by Latent Semantic Analysis. In *Journal of the American Society of Information Science*

Gale, W., K. Church, and D. Yarowsky. 1992. One Sense Per Discourse. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop*.

Goodman, J. 2003. Exponential Priors for Maximum Entropy Models.

Landauer, T. K., & Dumais, S. T. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104, 211-240, 1997.

MUC-7. 1998. Proceedings of the Seventh Message Understanding Conference.

Neal, R. M. 1993. Probabilistic Inference Using Markov Chain Monte Carlo Methods. Technical Report, Univ. of Toronto.

Pietra, S. D., V. D. Pietra, and J. Lafferty. 1995. Inducing Features Of Random Fields. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Srihari, R. K., W. Li, C. Niu and T. Cornell. InfoXtract: An Information Discovery Engine Supported by New Levels of Information Extraction. In *Proceeding of HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems*, Edmonton, Canada.